2009

COSMA – EIN INTEGRIERTER ANSATZ FÜR DAS MANAGEMENT VON SERVICE LEVEL AGREEMENTS BEI COMPOSITE SERVICES

André Ludwig
Universität Leipzig

COSMA – EIN INTEGRIERTER ANSATZ FÜR DAS MANAGEMENT VON SERVICE LEVEL AGREEMENTS BEI COMPOSITE SERVICES

André Ludwig[1]

Kurzfassung
Die Service-Bereitstellung in Service-Oriented Computing Umgebungen wird durch Service Level Agreements (SLA) beschrieben. Aktuelle SLA-Management-Ansätze konzentrieren sich auf bilaterale Service-Provider-/-Requester-Konstellationen, gehen jedoch nicht auf SLA-Management-Anforderungen von Composite Service Providern ein. Diese betreffen das SLA-Management mit Providern von Atomic Services, mit Requestern von Composite Services und deren Abstimmung aufeinander. Ein Composite SLA Management Ansatz muss den Anteil von (Atomic) SLAs beim SLA-Management von (Composite) SLAs berücksichtigen. Der COSMA-Ansatz erlaubt ein integriertes Management von Atomic und Composite SLAs während deren gesamten Lebenszyklus. Er ermöglicht die Abbildung der Abhängigkeiten und Beziehungen zwischen unterschiedlichen SLAs und damit eine Kontrolle, Steuerung und Optimierung der SLA-Management-Aktivitäten.

1. Einleitung

Service-Oriented Computing (SOC) beschreibt einen relativ neuen Ansatz für die Gestaltung von verteilten, unternehmensübergreifenden Informationssystemen. Mit SOC lassen sich Funktionalitäten von Informationssystemen als autonome, in sich geschlossene, Plattform-unabhängige Services abstrahieren, welche aufgrund ihrer Geschlossenheit wiederum in komplexe zusammengesetzte Services integriert werden können. Man unterscheidet dabei konzeptionell in *Atomic Services* zur Abstraktion einer bestimmten Funktionalität und *Composite Services*, welche entsprechend einem Prozessfluss aus Atomic Services gebildet werden können.

Eine Vision, die sich mit dem Gestaltungsansatz von SOC verbindet, ist die Etablierung einer global verfügbaren Infrastruktur, eines Service-Marktes für die Bereitstellung und Nachfrage von Services [12][6]. Diese Infrastruktur wird es Anbietern von Services (Service Provider) erlauben, zahlreiche Services in unterschiedlichen Service-Implementierungen mit individuellen Service-Eigenschaften einer Vielzahl von unterschiedlichen Service-Nachfragern (Service Requester) anzubieten, welche diese Service-Implementierungen zur Laufzeit (on-demand) an die nutzenden Anwendungen binden (dynamisches Binden).

[1] Universität Leipzig, Germany

Die Entstehung von Service-Märkten auf der Basis von SOC wird zur Etablierung des Geschäftsmodells des *Composite Service Providers (CSP)* führen [2][5]. Ein CSP fragt Atomic Services von externen Service Providern nach und bietet die durch einen bestimmten Prozessfluss verbundenen Atomic Services als in sich geschlossene Composite Services eigenen Kunden im Service-Markt an. Das heißt, ein CSP agiert als unabhängige Geschäftseinheit, die eigenen Interessen folgt, Profitabilität und durch zuverlässige Service-Bereitstellung Kundenzufriedenheit sicherstellen muss.

Die Definition, Steuerung und Kontrolle der Service-Bereitstellung kann durch Dienstgüteverträge, sogenannte *Service Level Agreements* (SLA), umgesetzt werden. SLAs bieten eine formale Beschreibung des ausgetauschten Services und definieren die Rechte und Pflichten der involvierten Parteien. Aktuelle SLA-Management-Ansätze für SOC-Umgebungen wie WSLA [8], WS-Agreement [3] oder WSOL [13] stellen umfangreiche SLA-Sprachformalisierungen und SLA-Management-Ausführungsumgebungen zur Verfügungen. Sie konzentrieren sich dabei jedoch auf bilaterale Service Requester-/Service Provider-Konstellationen und vernachlässigen die SLA-Management-Anforderungen von Composite Service Providern. Diese Anforderungen betreffen zum einen das SLA-Management mit Anbietern atomarer Service, das SLA-Management mit Nachfragern von Composite Services und zum anderen die Abstimmung beider SLA-Management-Aktivitäten aufeinander. Ein SLA-Management-Ansatz für Composite Services muss den Anteil extern bezogener Services – formalisiert in deren (Atomic) SLAs (*ASLA*) – beim SLA-Management der Composite Services – formalisiert in deren (Composite) SLAs (*CSLA*) – berücksichtigen.

Mit einer zunehmenden Anzahl von verfügbaren Service-Implementierungen, Service-Konfigurationen und Service-Anforderungen und den daraus resultierenden unterschiedlichsten Composite Service-Implementierungen erfordert (nahezu) jeder durch einen Composite Service Provider bereitgestellte Composite Service eine spezifische Anzahl von SLA-Dokumenten und ein individuelles Management dieser SLAs. Aufgrund des dynamischen Bindens und des dynamischen Bereitstellens von Services in Service-Märkten muss auch das Composite SLA-Management dynamisch zur Laufzeit und damit automatisch realisiert werden. Die manuelle Ausführung von SLA-Management-Aktivitäten, z.B. die Verhandlung und Erstellung von SLAs oder das Monitoring und die Evaluierung von SLAs, ist für das SLA-Management in Service-Märkten nicht ausreichend.

In diesem Beitrag wird ein neuer, integrierter Ansatz für das automatische Management von SLAs in Composite Services während ihres gesamten Lebenszyklus vorgestellt. Dieser Ansatz ist mit dem Akronym *COSMA* (*CO*mposite *S*la *MA*nagement) abgekürzt. COSMA ermöglicht die Abbildung von Beziehungen und Abhängigkeiten zwischen SLAs in Composite Services und erlaubt damit die zentrale Kontrolle, Steuerung und Optimierung der Composite SLA-Management-Aktivitäten bei einem CSP. Der Beitrag ist wie folgt strukturiert: Abschnitt 2 präsentiert die grundlegende Idee des COSMA-Ansatzes und stellt die konstitutiven Elemente von COSMA vor. Abschnitt 3 gibt einen kurzen Einblick in ein Anwendungsbeispiel und den COSMA Demonstrator und Abschnitt 4 fasst den Beitrag zusammen und gibt einen Ausblick in weitere Forschungsschritte.

2. Composite SLA Management-Ansatz (COSMA)

Die grundlegende Idee des COSMA-Ansatzes ist die Integration aller in einen Composite Service involvierten SLA-Dokumente in ein *Composite SLA Management Document*. Dieses Composite SLA Management Document, welches im weiteren als *COSMAdoc* bezeichnet wird, beinhaltet die vertraglichen Informationen, gekapselt in SLA-Dokumenten, und zusätzlich alle SLA-

Management-Daten insbesondere zur Abbildung der Beziehungen und Abhängigkeiten zwischen einzelnen SLA-Dokument-Elementen.

Auf der Basis Composite Service-spezifischer COSMAdoc-Instanzen kann der Beitrag von Atomic SLAs am Composite SLA abgebildet und nachvollzogen werden. Das SLA-Management-System eines CSP kann auf dieser Basis seine SLA-Management-Aktivitäten kontrollieren, steuern und optimieren. Dies betrifft insbesondere kompositionsweite SLA-Verhandlungen und kompositionsweite SLA-Monitorings und -Evaluierungen. Der COSMA-Ansatz besteht aus den folgenden Teilen:

- *COSMAdoc* stellt ein generisches Informationsmodell für die Integration von vertraglichen Daten und SLA-Management-Daten zur Verfügung. COSMAdoc erweitert dabei ein bestehendes SLA-Informationsmodell und stellt Elemente für die Abbildung von Beziehungen und Abhängigkeiten zwischen SLA-Elementen zur Verfügung.
- *COSMAframe* beschreibt ein konzeptionelles Rahmenwerk (Framework) für das dynamische Management von Composite SLAs. Das Rahmenwerk beschreibt dabei die generischen Komponenten sowie deren Nutzung von COSMAdoc-Instanzen zur Umsetzung der SLA-Lebenszyklus-Management-Aktivitäten.
- *COSMAlife* stellt eine Sammlung von Mechanismen und Protokollen zur Verfügung, die definieren, wie die Komponenten von COSMAframe die Managementaktivitäten des SLA Lebenszyklus (Lifecycle) auf Basis von COSMAdoc-Instanzen umsetzen.

COSMAframe, COSMAdoc und COSMAlife sind in die vorhandene Service Provisioning Infrastruktur des CSP eingebettet (z.B. SOA-Plattform, Enterprise Service Bus). Diese wird im Folgenden als Operation and Management System (OMS) bezeichnet. Das OMS stellt zahlreiche weitere Funktionen, beispielsweise zur Registrierung von Services, zur Interaktion mit Service Requestern und Providern und zur Administration, zur Verfügung. Weiterhin stellt das OMS Komponenten zur automatischen Erstellung von Composite Service-Orchestrierungsskripts (Service Composer), zur automatischen, Agenten-basierten Verhandlung von SLAs (Negotiation Engine) oder zur automatischen Ausführung und Kontrolle von Services (Service Enactment & Monitoring Engine) zur Verfügung[2].

2.1 Informationsmodell COSMAdoc

Das generische Informationsmodell COSMAdoc stellt das informationelle Herz des COSMA-Ansatzes dar. Es kapselt die vertraglichen Informationen der involvierten SLAs und deren SLA-Management-Informationen in einer zentralen Einheit. Dabei wird für jeden involvierten Atomic Service ein SLA-Dokument (ASLA) und zusätzlich für den Composite Service ein SLA-Dokument (CSLA) erstellt. Zusätzlich werden die Beziehungen und Abhängigkeiten zwischen Elementen dieser SLA-Dokumente in COSMAdoc gespeichert. Für jeden durch einen CSP bereitgestellten Composite Service wird eine spezifische Instanz von COSMAdoc erzeugt und an den bereitgestellten Service gebunden. Die COSMAdoc-Instanz wird für alle in COSMAlife definierten Aktivitäten durch die COSMAframe-Komponenten verwendet. Auf der Basis der spezifischen, in der COSMAdoc-Instanz gespeicherten Beziehungen und Abhängigkeiten können die Komponenten von COSMAframe beispielsweise kompositionsweite SLA-Verhandlungen und Überprüfungen der SLA-Einhaltung durchführen. Aufgrund der Zusammenführung von SLA- und SLA-Management-

[2] Die innerhalb des „Adaptive Services Grid“ (ASG) Projekts entwickelte ASG-Plattform stellt prototypische Implementierungen der vorgestellten Komponenten des OMS zur Verfügung. Weitere Informationen sind auf der Projekt-Website verfügbar [4].

Informationen in einem zentralen Dokument stellt eine COSMAdoc-Instanz ein zentrales Managementwerkzeug für einen CSP dar, welches CSP-intern zur Kontrolle, Steuerung und Optimierung der SLA-Management-Aktivitäten genutzt wird. Die eingebetteten SLA-Dokumente enthalten dabei ausschließlich vertragliche Daten, die den jeweils beteiligten Vertragsparteien zugänglich gemacht werden können.

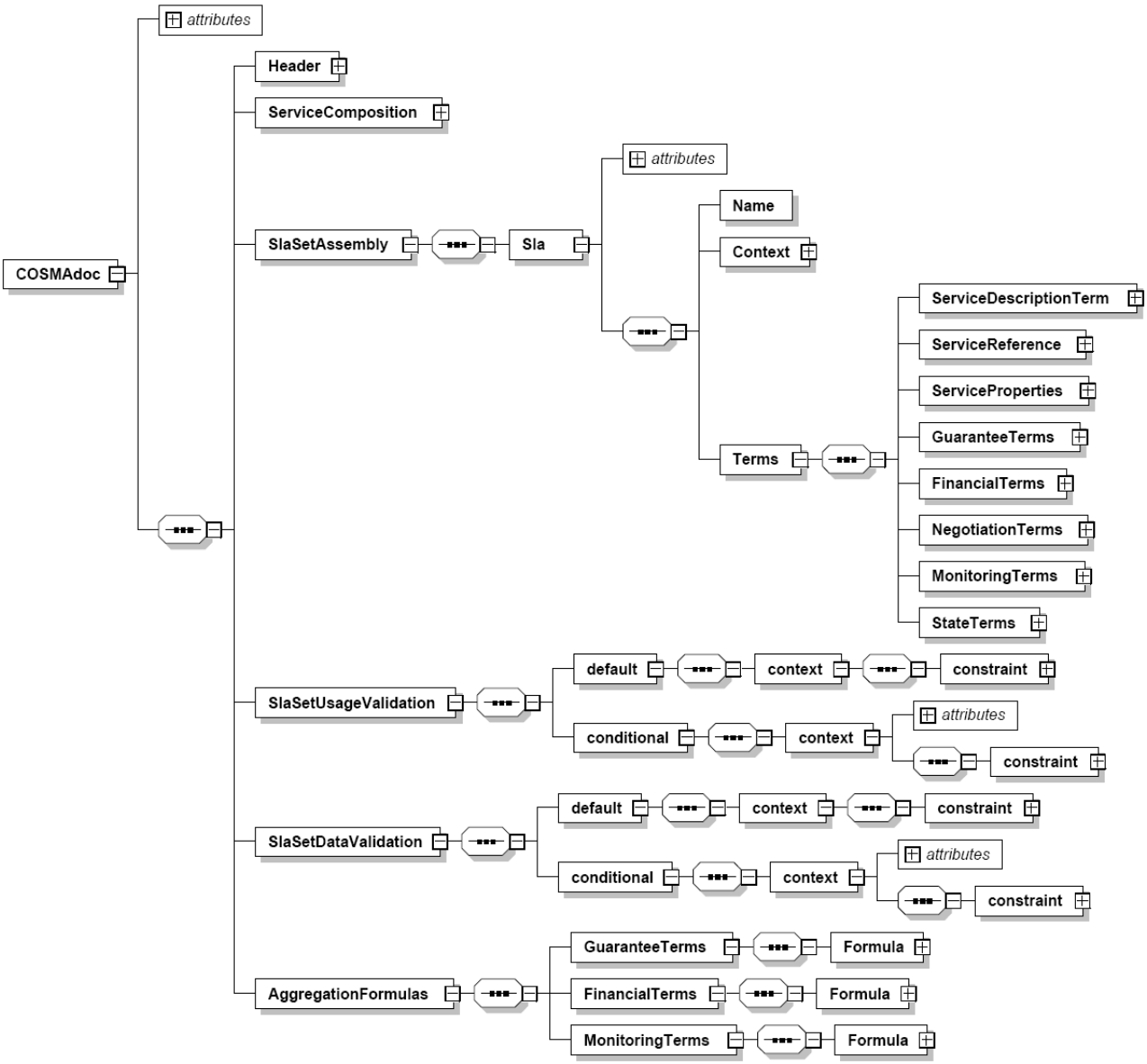


Abbildung 1: Struktur des COSMAdoc-Informationsmodells

COSMAdoc besteht aus sechs Kernelementen (vgl. Abbildung 1). Alle Kernelemente sind selbständige Teile, die teilweise über Referenzen miteinander verbunden sind. Die Elemente erfüllen die folgenden Aufgaben:

- *Header*: Der Header einer COSMAdoc-Instanz dient zur Definition von Kontextinformationen und Parametern. Beispielsweise lassen sich der Besitzer der Instanz, eine Versionsnummer oder Beschreibung definieren. Zusätzlich können die in der COSMAdoc-Instanz verwendeten semantischen Modelle referenziert werden.
- *ServiceComposition*: Das Element ServiceComposition erlaubt die Speicherung des dem Composite Service zugrunde liegenden Orchestrierungsskripts (z.B. in WS-BPEL [11]). Das

Orchestrierungsskript dient nicht nur der Ausführung des Composite Service, sondern bestimmt maßgeblich die Beziehungen zwischen ASLA und CSLA Elementen.

- *SlaSetAssembly*: In SlaSetAssembly werden *Sla*-Elemente zur Repräsentation von ASLAs und CSLA gespeichert. Dabei werden ausschließlich vertragliche Informationen in Sla-Elementen gespeichert. Die Struktur jedes Sla-Elements basiert auf dem dreiteiligen SLA-Modell der WS-Agreement Spezifikation [3]. Die vom Open-Grid-Forum (OGF) erarbeitete Spezifikation bietet eine umfangreiche Grundlage zur Formalisierung von bilateralen Verträgen zwischen Vertragspartien. Das COSMAdoc SLA-Modell erweitert das WS-Agreement SLA-Modell um MonitoringTerms zur Definition der Monitoring-Konfiguration (vgl. z.B. Ausführungen in [8]), NegotiationTerms zur Speicherung des zugrunde liegenden Verhandlungsprotokolls und FinancialTerms zur strukturierten Speicherung von finanziellen Aspekten des SLA (z.B. Verrechnungsmodelle und -entgelte, Sanktionen und Boni).
- *SlaSetUsageValidation*: Der Bereich SlaSetUsageValidation ermöglicht die Kontrolle und Restriktion der *Verwendung* von Sla-Element-Teilen der SlaSetAssembly. Beispielweise lassen sich Teile von Sla-Elementen als obligatorisch, optional oder verhandelbar deklarieren. Weiterhin können Formatrestriktionen, z.B. bestimmte Datumsformate oder Nachkommastellen, zugewiesen werden. Restriktionen, welche die Verwendung einschränken, lassen sich als immer geltende (default) und bei Erfüllung einer bestimmten Bedingung geltend (conditional) definieren. Zur Identifizierung der Sla-Element-Teile werden Pointer verwendet (z.B. durch Anwendung der Syntax von XPath [14]).
- *SlaSetDataValidation*: Während die SlaSetUsageValidation die Verwendung von Sla-Elementen kontrolliert, können mit dem Bereich SlaSetDataValidation die *Daten* der beinhalteten Sla-Elemente kontrolliert werden. Durch Prädikate lassen sich Teile von Sla-Elementen in ihren zulässigen Maximal-/Minimalwerten, Wertebereichen oder auf feste Werte beschränken. Da insbesondere die Daten von CSLA-Elementen von Daten der ASLA-Elemente abhängen, lassen sich die Prädikate der SlaSetDataValidation an Aggregationsformeln knüpfen, welche die Daten der Sla-Elemente unterschiedlicher SLAs über Algorithmen miteinander verbinden. Durch die Verknüpfung von Prädikaten mit Aggregationsformeln kann gewissermaßen eine Querverbindung zwischen SLA-Elementen unterschiedlicher Sla‘s dargestellt werden. Die Verwendung von Aggregationsformeln ist dennoch nicht auf das Zusammenführen von ASLA- und CSLA-Elementen beschränkt, sondern lässt sich auch auf Zusammenhänge zwischen ASLA-Elementen anwenden. Abbildung 2 veranschaulicht die Zuweisung von Aggregationsformeln aus dm Bereich der *AggregationFormulas* über Prädikate der SlaSetDataValidation an Teilen von Sla-Elementen (z.B. an Service Level Objectives in GuaranteeTerms). Ähnlich wie bei SlaSetUsageValidation lassen sich auch im Bereich SlaSetDataValidation Prädikate als default oder conditional definieren.
- *AggregationFormulas*: Im Bereich der AggregationFormulas werden die Beziehungen und Abhängigkeiten zwischen Elementen unterschiedlicher SLAs in Form von Aggregationsformeln (Formula) gespeichert. Die Aggregationsformeln können unterschiedlichste mathematische Operatoren verwenden und referenzieren die verwendeten Terme (Teile von Sla-Elementen aus dem Bereich der SlaSetAssembly) durch Pointer (z.B. unter Verwendung der XPath Syntax [14]). Aggregationsformeln werden dabei für GuaranteeTerms, FinancialTerms und MonitoringTerms (Teile des COSMAdoc SLA-Modells) gespeichert. Die Aggregationsformeln und deren Komplexität hängen im Wesentlichen von zwei Faktoren ab: der Struktur des Orchestrierungsskripts und der Art und Anzahl der in den SLAs verwendeten SLA-Parametern (z.B. Verfügbarkeit, Antwortzeit, Verrechnungsentgelt). Zur Erzeugung der Aggregationsformeln müssen

zunächst die unter Umständen komplexen, geschachtelten Orchestrierungsskripte in ihre atomaren Kompositionsmuster (z.B. Sequenz, Schleife, Parallel Split) zerlegt werden und anschließend Aggregationsformeln für verschiedene SLA-Parameter erstellt werden. Etablierte Ansätze, die bei der Erzeugung von Aggregationsformeln Verwendung finden können, stellen beispielsweise [7] und [15] dar.

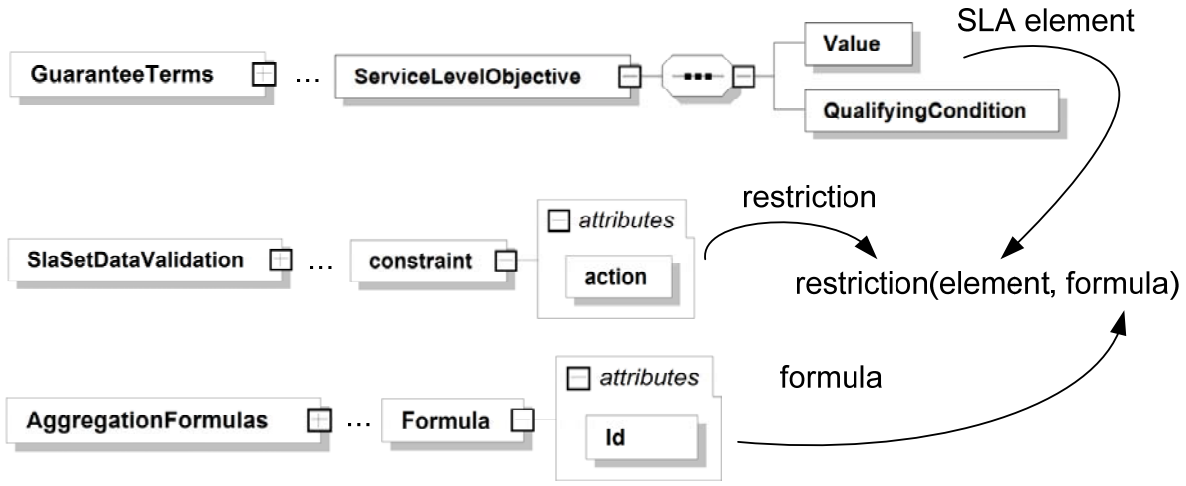


Abbildung 2: Zuweisung von Aggregationsformeln über Prädikate an SLA Elementen

Die Kernelemente von COSMAdoc bilden eine feste Struktur und erlauben die Definition eines generischen COSMAdoc-Templates. Dieses Template ist erweiterbar und kann an unterschiedliche Anwendungsbereiche angepasst werden. Für unterschiedliche von einem CSP angebotene Composite Services werden individuelle Dokumentinstanzen des COSMAdoc-Templates gebildet.

2.2 Konzeptionelles Rahmenwerk COSMAframe

COSMAframe stellt ein konzeptionelles Rahmenwerk dar, welches die generischen Komponenten, die für ein automatisches Verarbeiten von COSMAdoc-Instanzen, z.B. SLA-Erstellung und -Verhandlung oder SLA-Monitoring und -Evaluierung notwendig sind, definiert. COSMAframe ist als abstrakte Lösung zu interpretieren, in der die Komponenten in ihrem grundlegenden Verhalten und ihren generischen Schnittstellen beschrieben werden. COSMAframe ist kein Implementierungsrahmenwerk und kein Rahmenwerk, welches feste Programmierschnittstellen und Datenobjekte beschreibt, sondern ist notwendigerweise zu erweitern und anzupassen. COSMAframe besteht aus den folgenden generischen Komponenten (vgl. Abbildung 3):

- *COSMA Manager*: Der COSMA Manager stellt die zentrale Steuerungs- und Verwaltungskomponente in COSMAframe dar, die alle anderen Komponenten entsprechend der notwendigen Management-Aktivitäten im SLA-Lebenszyklus triggert. Die Komponente bietet einen zentralen Port für externe Komponenten und stellt eine Validierungsschnittstelle bereit, welche die SlaSetUsageValidation und SlaSetDataValidation Prädikate in COSMAdoc-Instanzen auswertet und damit beispielsweise durch Verhandlungsmaschinen (Negotiation Engines) nutzbar macht.
- *COSMAdoc Creator*: Die Komponente COSMAdoc Creator ist für die kompositionsspezifische Erzeugung von COSMAdoc-Instanzen verantwortlich. Auf der Basis eines durch einen Service Composer erstellten Orchestrierungsskripts modularisiert die Komponente die Komposition und baut die COSMAdoc-Instanz aus dem generischen COSMAdoc Template auf.
- *COSMAdoc Integrator*: Die kompositionsspezifische COSMAdoc-Instanz wird im COSMAdoc Integrator um weitere Elemente ergänzt. Der COSMAdoc Integrator fügt

zunächst sämtlich SLA-Elemente, wie Servicequalitäts-Parameter und deren Metriken, finanzielle Parameter wie Entgelt-/Sanktionen-/Bonus-Verrechnungsmodelle und die Monitoring- und Verhandlungskonfigurationen ein. Anschließend werden alle direkt ableitbaren SlaSetUsageValidation- und SlaSetDataValidation-Prädikate erstellt und zusammen mit den dazugehörigen Aggregationsformeln in die COSMAdoc-Instanz eingebettet. Zum Teil werden dazu Konfigurationen verwendet, welche in einer Datenbank hinterlegt sind.

- *COSMAdoc Repository*: Die erstellten und integrierten COSMAdoc-Instanzen werden im COSMAdoc Repository gespeichert. Das Repository ist für alle internen COSMAdoc-Instanzen erreichbar.
- *COSMAdoc Validator and Violation Detector*: Zur Validierung und Erkennung von Verletzungen der in einen Composite Service involvierten SLAs wird die Komponente COSMAdoc Validator and Violation Detector verwendet. Die Komponente vergleicht Monitoring-Messergebnisse zur Laufzeit mit in den ASLAs verwendeten Service Level Objectives (Mindest-Service Levels). Auf dieser Basis und unter Hinzuziehung der gespeicherten Aggregationsformeln wird ermittelt, ob Verletzungen bei ASLAs zur Verletzung der CSLA führen. Es wird damit dokumentiert, welcher Atomic Service für eine Verletzung der CSLA verantwortlich ist. Die Komponente erstellt auf dieser Basis Vorschläge zur Behandlung der Verletzung, z.B. Forderung von Sanktionen durch die Verursacher bzw. Zahlung von Sanktion an den Nachfrager des Composite Service, und stellt diese Vorschläge dem COSMA Manager zur Verfügung.

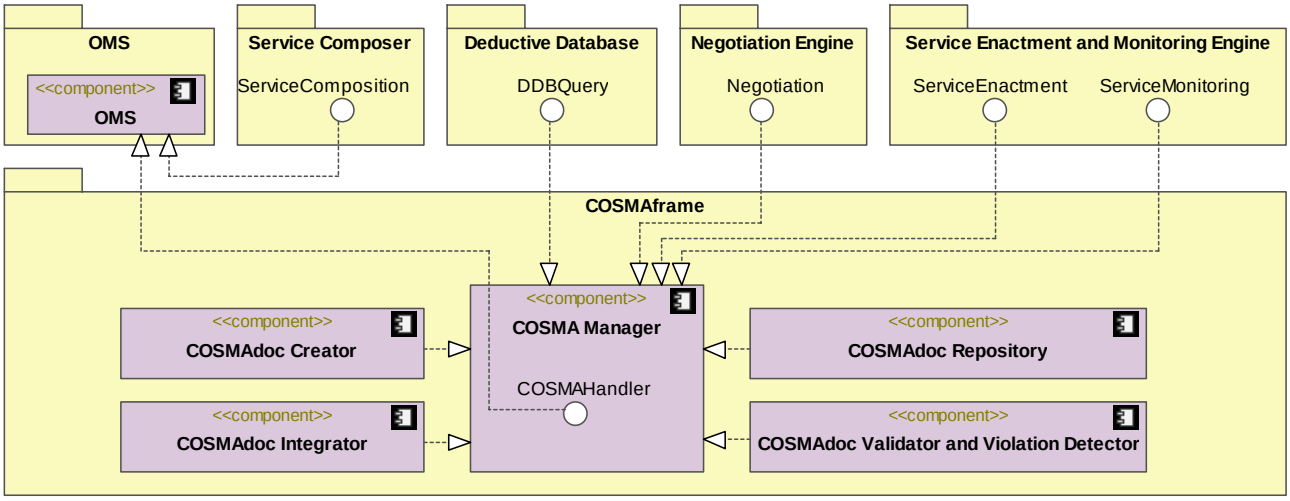


Abbildung 3: COSMAframe Komponentenüberblick sowie externe Komponenten

Die Komponenten von COSMAframe nutzen eine Reihe externer Komponenten. Dazu zählen beispielsweise ein Service Composer zur automatischen Erstellung von generischen Orchestrierungsskripten zur Laufzeit, eine Negotiation Engine zur automatischen Verhandlung von SLAs durch autonome Verhandlungsagenten, eine Service Enactment and Monitoring Engine zur schrittweisen Ausführung und zum Monitoring der Atomic und Composite Services. Diese Komponenten zählen nicht zu COSMAframe und es muss auf existierende Ansätze und Prototypen zurückgegriffen werden (vgl. z.B. prototypische Komponentenimplementierungen im Adaptive Services Grid Projekt [4]).

2.3 Mechanismen zum Management des Composite SLA Lebenszyklus COSMAlife

COSMAlife stellt einen integrierten Satz von Mechanismen für das Management des SLA-Lebenszyklus bereit. Dabei fokussiert COSMAlife insbesondere die für einen CSP besonders relevanten Phasen: SLA-Erstellung und -Verhandlung und SLA-Monitoring und -Evaluierung.
Die Phase des Durchsetzens von SLA-Festlegungen durch die Service-Bereitstellungsinfrastruktur (SLA Provisioning & Fulfilment) ist für das Grundmodell des CSP irrelevant, da er ausschließlich externe Atomic Services integriert. Weitere Phasen des SLA-Lebenszyklus wie SLA-Terminierung und -Erklärung werden in diesem Beitrag nicht näher betrachtet.

Generell kann das Management von SLAs als zyklischer Prozess mit iterativen Managementschritten charakterisiert werden. Das heißt, die SLA-Lebenszyklus-Management-Aktivitäten müssen für jeden durch den CSP bereitgestellten Composite Service parallel ausgeführt werden. Der aktuelle Status innerhalb des SLA-Lebenszyklus wird durch die in COSMAdoc enthaltenen StateTerms gespeichert. Aus Platzgründen soll an dieser Stelle auf nähere Ausführungen zu COSMAlife verzichtet werden. COSMAlife Protokolle zur Verhandlung von SLAs in Composite Services werden in [9] vorgestellt.

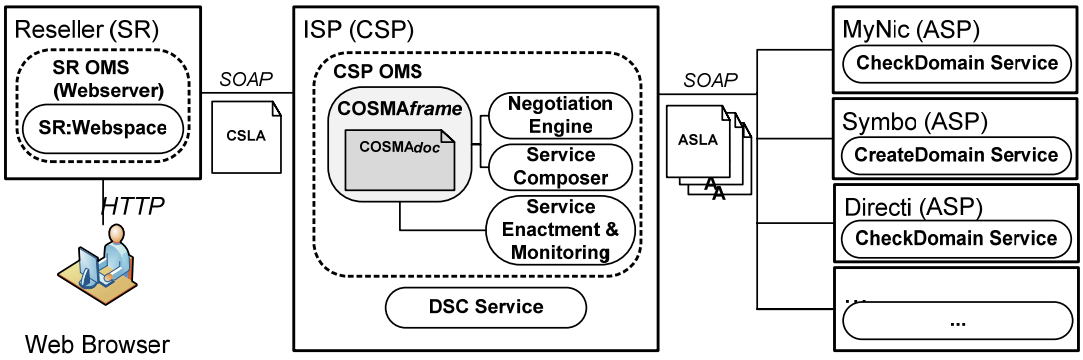
3. Anwendungsszenario und Demonstrator

Zur Illustration der Anwendung des COSMA-Ansatzes und seiner konstitutiven Teile COSMAdoc, COSMAframe und COSMAlife wurde ein Anwendungsszenario entwickelt, welches auf einem mit Partnern aus Industrie und Wissenschaft entwickelten Anwendungsszenario aus dem Adaptive Services Grid EU-Projekt aufbaut [10]. Das Szenario beschreibt einen Internet Service Provider (ISP), der als CSP unterschiedliche, Kunden-individuell zusammengesetzte Services im Bereich Webspace-Provisioning anbietet. Zu den bezogenen Atomic Services zählen Services zur Registrierung von Domain Names, zur Abwicklung von Kreditkartenzahlungen oder zum Setup des Webspace Hostings. Der angebotene Composite Service fasst diese Atomic Services entsprechend einem Orchestrierungsskript zusammen und wird unterschiedlichen Service Requestern angeboten. Im Anwendungsszenario wird gezeigt, wie der ISP eine auf COSMA basierende Composite SLA Management-Lösung zur Realisierung der SLA-Management-Aktivitäten nutzt.

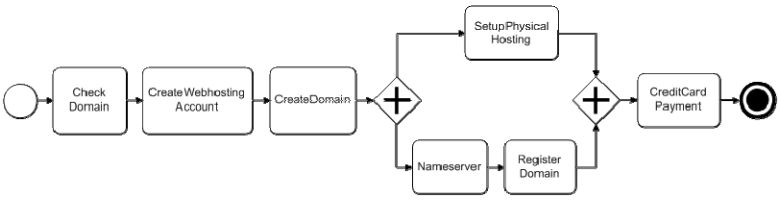
Zusätzlich wurde ein Demonstrator entwickelt, der die Protokolle und Mechanismen von COSMAlife umsetzt, individuelle COSMAdoc-Instanzen erzeugt und zum SLA-Management verwendet und damit eine stabile Grundlage zur explorativen weiteren Untersuchung des COSMA-Ansatzes bietet. Mit dem Demonstrator und den hinterlegten Testfällen können Experimente und eine schrittweise Überprüfung der kompositionsweiten SLA-Verhandlungen bzw. -Monitorings durchgeführt werden. Sowohl das Anwendungsszenario als auch der Demonstrator sind als repräsentative Umsetzungen des COSMA-Ansatzes zu verstehen und in keiner Weise auf diese beschränkt.

Abbildung **4** gibt einen Überblick über das Anwendungsszenario, den verwendeten Composite Service und zeigt einen Screenshot des COSMA-Demonstrators.

Übersicht Anwendungsszenario:



Composite Service:



COSMA Demonstrator:

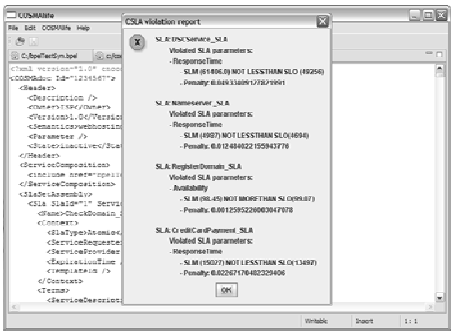


Abbildung 4: Übersicht Anwendungsszenario, Composite Service und COSMA-Demonstrator

4. Zusammenfassung und Ausblick

Das Thema Composite SLA Management in SOC-Umgebungen ist ein relativ junges Forschungsthema und insbesondere durch die hohe Dynamik in SOC-Umgebungen und der damit verbundenen Anforderung zur Automatisierung charakterisiert. COSMA stellt einen neuen Ansatz für das automatische Management von SLAs bei Composite Services bereit und ermöglicht einem CSP damit seine SLA-Management-Aktivitäten automatisch zu kontrollieren, zu steuern und zu optimieren. Dies betrifft die Maximierung der Profitabilität, indem SLA-Datenrestriktionen als Basis für SLA-Verhandlungen und Berechnungen von finanziellen Aspekten verwendet werden können (Minimierung der Kosten für bezogene Services und Maximierung des Composite Service Verkaufspreises). Weiterhin kann über die Kontrolle der SLA-Daten der Qualitätsbeitrag von Atomic Services zu Composite Services abgebildet und damit zusicherbare Servicequalität definiert werden (Minimierung von notwendigen Servicequalitätsanforderungen zu Atomic Services bei gleichzeitiger Erreichung der geforderten Servicequalität für Composite Services).

COSMA stellt einen konzeptionellen Ansatz dar, der entsprechend individuellen Anforderungen und domänenspezifischen Besonderheiten angepasst und erweitert werden muss. Untersuchungen von COSMA mit weiteren Anwendungsszenarien und einer größeren Anzahl von Services in unterschiedlichen Service-Implementierungen wird zu weiteren Erkenntnissen und Anpassungen des Ansatzes führen. Darin ist auch eine Ausweitung von COSMAlife auf die Phase SLA-Terminierung und Voraussage von Verletzungen einbegriffen. Nach der Terminierung von SLAs in COSMAdoc-Instanzen sollte das Service-Verhalten in dynamischen Service Profilen (vgl. z.B. DSP-Ansatz aus [1]) gespeichert und bei der Erzeugung von Daten- und Verwendungsrestriktionen (SlaSetDataValidation bzw. SlaSetUsageValidation) einbezogen werden. Dadurch lässt sich der Prozess der SLA-Verhandlungen iterativ verbessern und zusätzlich unterschiedliche SLA-Lebenszyklen miteinander verbinden.

Quellen

[1] ABRAMOWICZ, W., KACZMAREK, M., KOWALKIEWICZ, M., ZYSKOWSKI, D., Architecture for Service Profiling. In: Proceedings of the Services Computing Workshops SCW'06, Chicago 2006, S. 121-130.

[2] ALONSO, G., CASATI, F., KUNO, H., MACHIRAJU, V., Web services: concepts, architectures and applications. Springer, Berlin, New York 2004.

[3] ANDRIEUX, A., CZAJKOWSKI, K., DAN, A., KEAHEY, K., LUDWIG, H., NAKATA, T., PRUYNE, J., ROFRANO, J., TUECKE, S., XU, M., Web Service Agreement Specification (WS-Agreement). http://www.gridforum.org/documents/GFD.107.pdf, letzter Zugriff am 04.06.2008.

[4] ASG: Integrated Project Adaptive Services Grid, ASG Projekt-Website: http://www.asg-platform.org, letzter Zugriff am 13.06.2008.

[5] BENATALLAH, B., DUMAS, M., MAARMAR, Z., Definition and Execution of Composite Web Services: The SELF-SERV Project. In: Data Engineering Bulletin, 25 (2002) 4, S. 1-6.

[6] HEUSER, L., The Internet of Services. http://www.fit.qut.edu.au/industry/corporateevents/docs/22Oct07_ProfessorHeuser_SAPResearch.pdf, letzter Zugriff am 02.03.2008.

[7] JAEGER, M.C., ROJEC-GOLDMANN, G., MUEHL, G., QoS aggregation for Web service composition using workflow patterns. In: Proceedings of the The 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC 2004), Monterey 2004, S. 149-159.

[8] KELLER, A., LUDWIG, H., The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. In: Journal of Network and Systems Management, Special Issue on E-Business Management, 11 (2003) 1, S. 57-81.

[9] LUDWIG, A. FRANCZYK, B., Managing Dynamics of Composite Service Level Agreements with COSMA. Akzeptiert zur Publikation bei 5th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2008), Jinan, China, 18.-20.10.2008.

[10] MOMOTKO, M., GAJEWSKI, M., LUDWIG, A., KOWALCZYK, R., KOWALKIEWICZ, M., ZHANG, J.Y., Towards adaptive management of QoS-aware service compositions. In: Multiagent and Grid Systems, 3 (2007) 3, S. 299-312.

[11] OASIS: Organization for the Advancement of Structured Information Standards, Web Services Business Process Execution Language Version 2.0 (WS-BPEL). http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html, letzter Zugriff am 23.11.2007.

[12] PAPAZOGLOU, M.P., Web Services: Principles and Technology. Prentice Hall, Essex 2007.

[13] TOSIC, V., PATEL, K., PAGUREK, B., WSOL - Web Service Offerings Language In: Proceedings of the International Workshop on Web Services, E-Business, and the Semantic Web (WES2002), Toronto 2002, S. 57-67.

[14] W3C: World Wide Web Consortium, XML Path Language (XPath). http://www.w3.org/TR/xpath, letzter Zugriff am 26.11.2007.

[15] ZENG, L., BENATALLAH, B., NGU, A.H.H., DUMAS, M., KALAGNANAM, J., CHANG, H., QoS-Aware Middleware for Web Services Composition. In: IEEE Transactions on Software Engineering, 30 (2004) 5, S. 311-327.